

# Project Write-Up

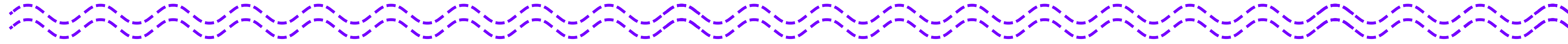


Ruslan Mukhamedvaleev

**UW CSE 455**  
**Advanced Vision**

The writing in this PDF is my original work for the Kernel Demo tool. The project can be found entirely on GitHub (<https://github.com/DerekWWang/KernelCooker>) and is also hosted on the web (<https://kernel-cooker.vercel.app/>).

# Goal



I was invited by a University of Washington student, Derek Zhu, to create a tool for visually demonstrating “image kernels” for students taking Advanced Vision (CSE 455). At the time, I had no clue what image kernels were or how we were going to visualize them, but I was excited about learning and taking on the challenge!

---

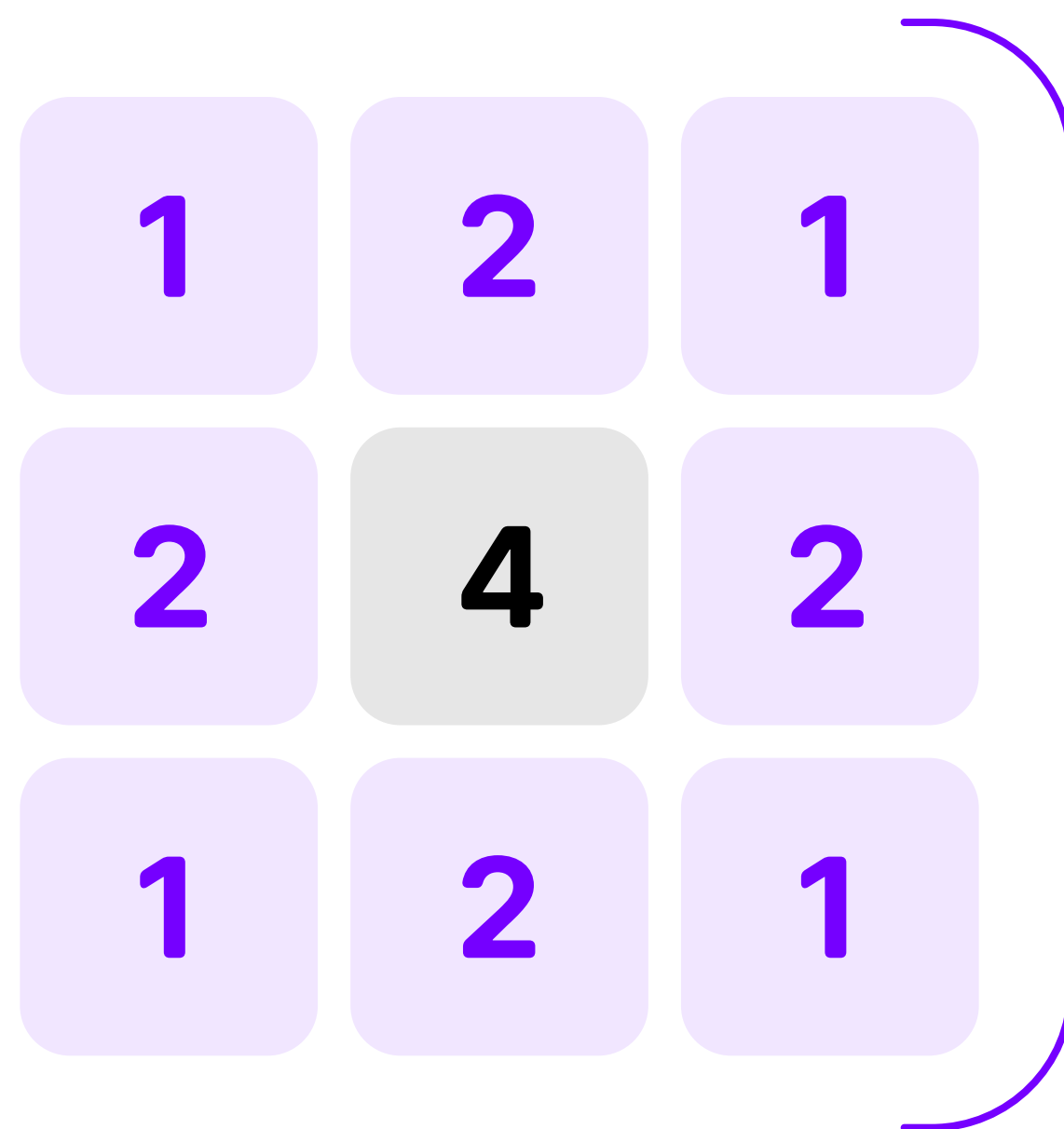
**Goal #1:** Learn about kernels within image processing

**Goal #2:** Create a useful tool for visually representing the concept

---

# Image Kernels

What is an image kernel? Well, it's actually just a 2D matrix that is applied to the pixels of an image for effects like blurring, sharpening, and many more. Matrices are used because images are matrices themselves.



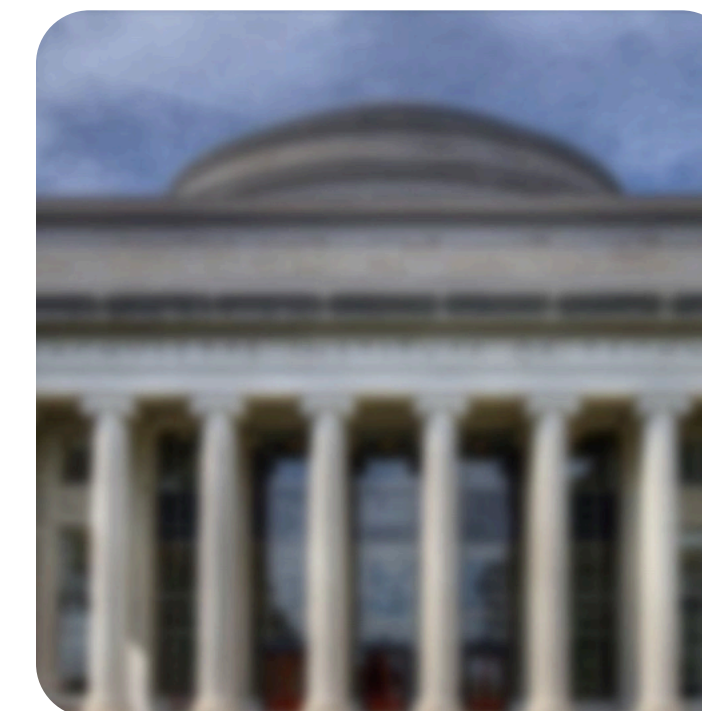
## Matrix for Blur

The 3×3 matrix is applied to each pixel of the image. The main pixel is marked in black, with values getting pulled from its neighbors. A blur matrix, like this, tries to decrease the differences between the main pixel and its neighbors. Sharpness does the opposite.

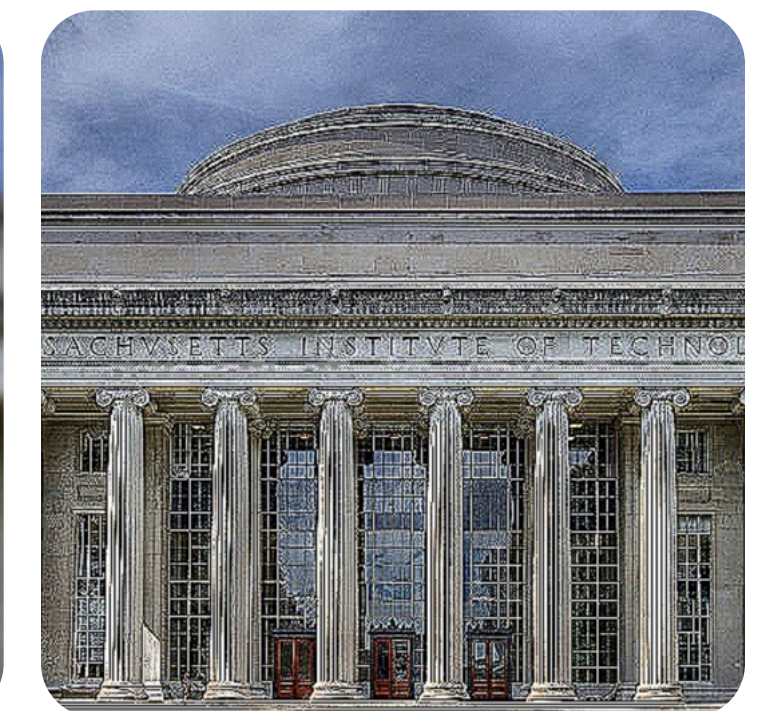
## Normal



## Blur

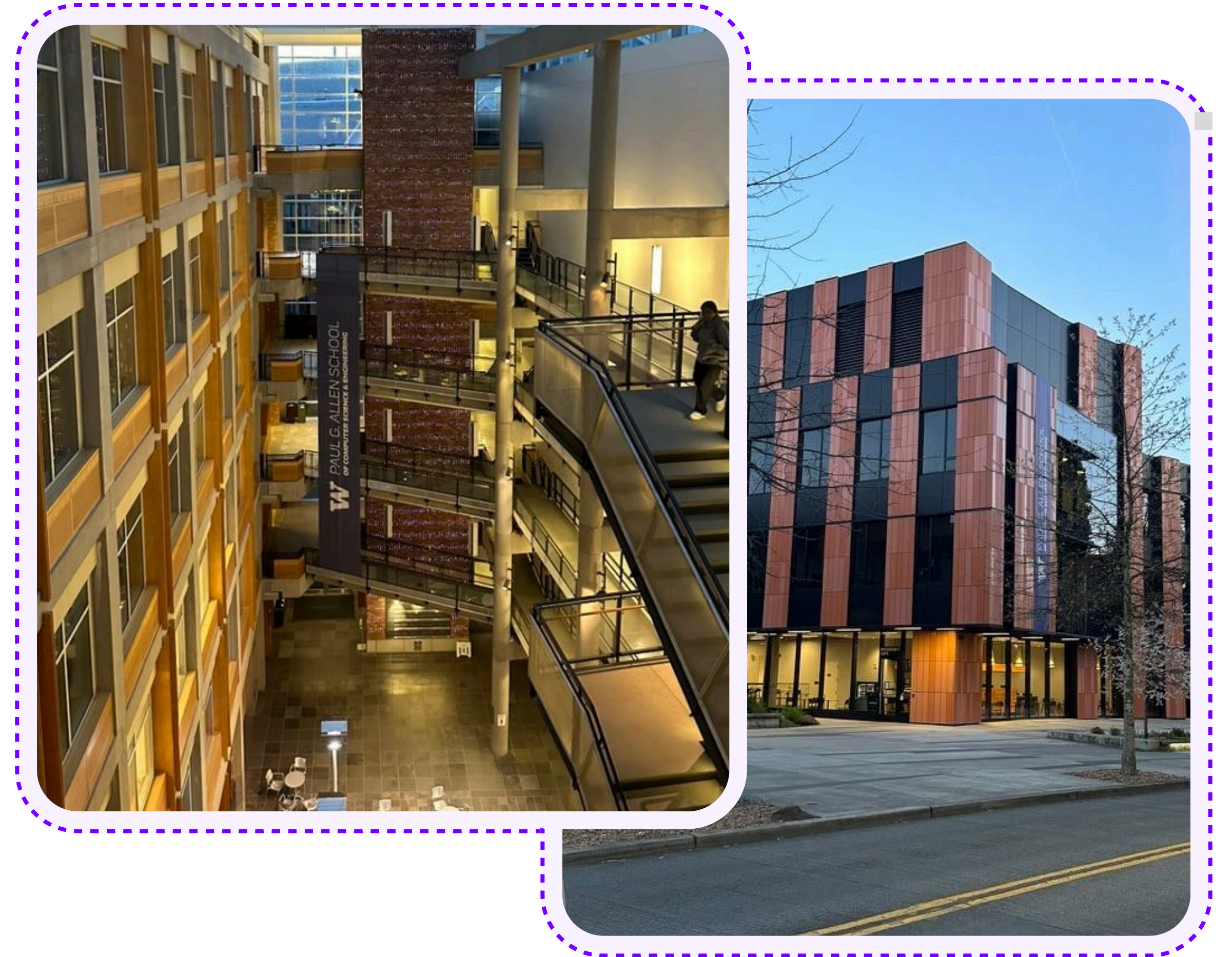


## Sharpen

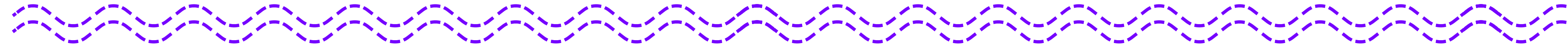


# Team

- Derek and I built this project over the course of about a week in March 2024. I was invited to work on this project by him because of my UI/UX and web programming skills.
- He worked on the foundational math and the application of the kernels passed into the images. I was in charge of the frontend and helped add fraction calculations (for more complex kernels that use fractions, like Gaussian blur).



# Planning

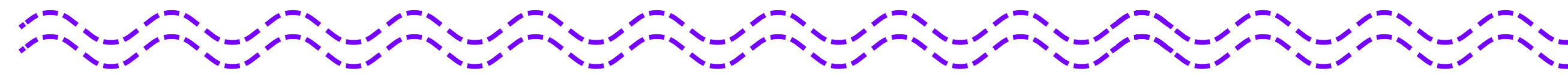


After discussing things at more depth, I arrived at three important things that this tool had to balance (in thinking about user experience), which other ones did not prioritize for it to actually be successful in helping the class:

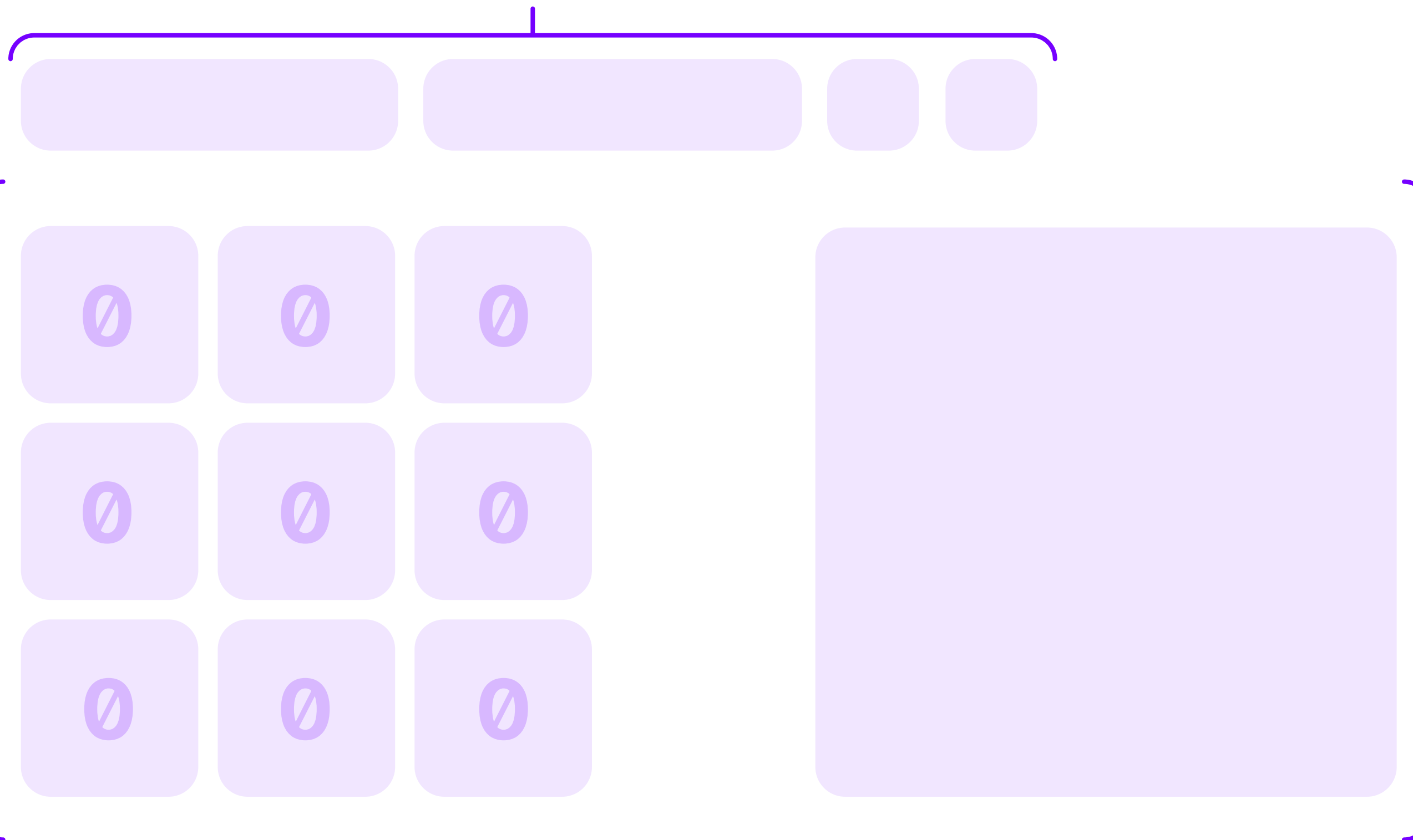
1. A modern, easily usable interface that abstracted enough away to make it intuitive.
2. Accessible from a variety of devices that students would have (no installing things).
3. Open source, so that students can build off of the project as a foundation.

All three are important and centrally revolve around accessibility. I knew this tool needed to be able to grow and change, even after we were done building it.

# Planning the Interface



Control Panel Common images & kernels w/ the ability to add custom ones



## 3x3 Matrix Input

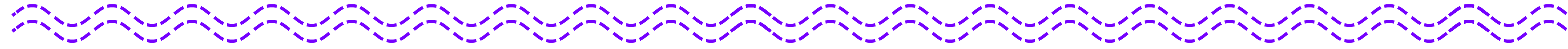
3x3 image kernels are the most common, and all we really need the tool to be able to demo.

We or another student can later expand upon this! We also want the ability to add custom fractions for more complex kernels.

## Processed Image

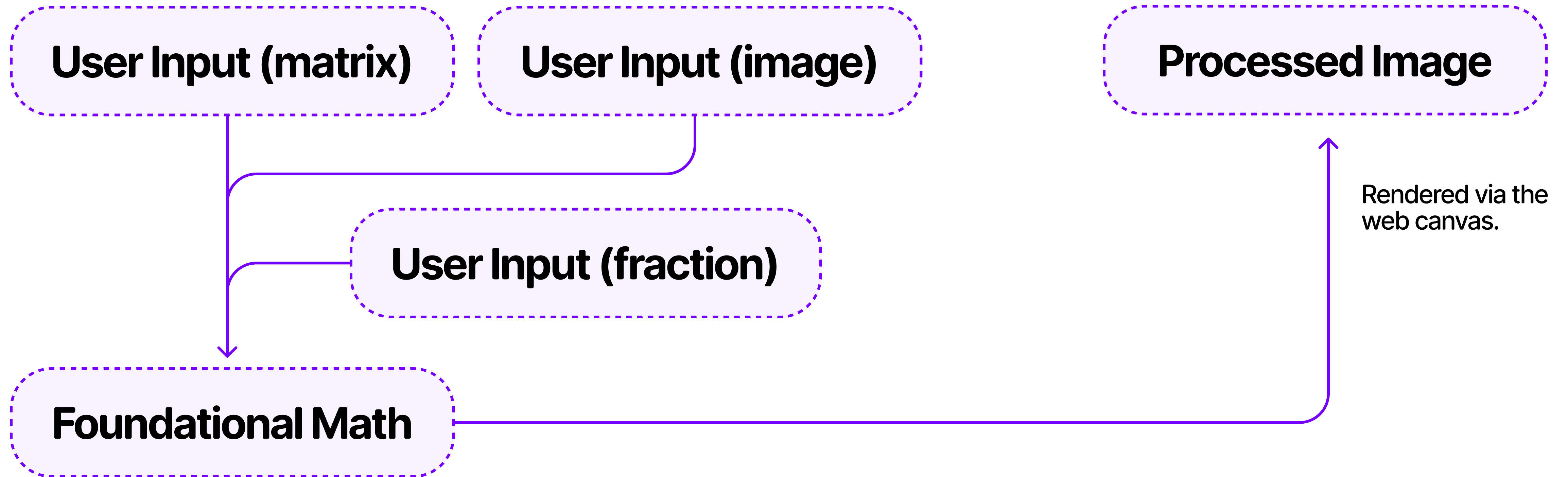
The final image would be displayed back to the user in a way that allows them to iterate on their changes (maybe by combining multiple kernels).

# Building



- We decided to go with a web app to make it as accessible on as many devices as possible. For our stack, we chose Vite.js (a React framework), as both of us had experience with the framework as well as JavaScript and JSX.
- There was no traditional backend for this project, as everything was done with in-browser processing. MathJS, a huge math library for JavaScript, powered our matrices.

# Data Flow

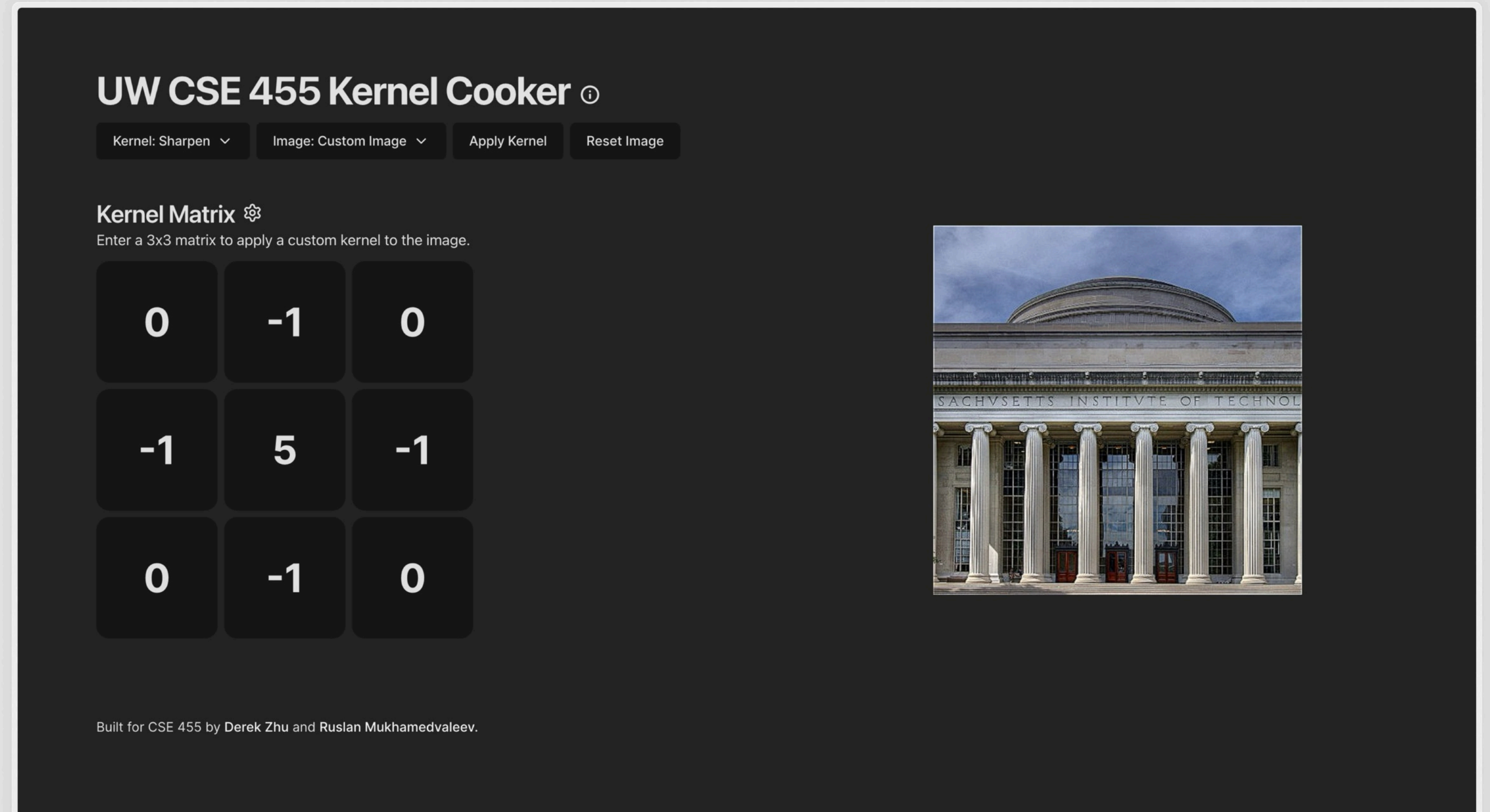


All processing (math) is done locally right in the browser, so there is no “backend” for this project!

# Tool

This is our final tool! You are able to select from four base images and seven base kernels, with the ability to add custom images and kernels quickly. You can also customize the fraction multiplier used by some of the kernels. I followed my initial draft closely!

View the attached video demo to see a better visual or visit <https://kernel-cooker.vercel.app/> to play around with it!




### UW CSE 455 Kernel Cooker ⓘ

Kernel: Gaussian Blur ▾ Image: Gates Building ▾ Apply Kernel Reset Image

Kernel Matrix ⓘ Fraction: 1/16  
Enter a 3x3 matrix to apply a custom kernel to the image.

|   |   |   |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |




Built for CSE 455 by Derek Zhu and Ruslan Mukhamedvaleev.

### UW CSE 455 Kernel Cooker ⓘ

Kernel: Ridge ▾ Image: Custom Image ▾ Apply Kernel Reset Image

Kernel Matrix ⓘ  
Enter a 3x3 matrix to apply a custom kernel to the image.

|    |    |    |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8  | -1 |
| -1 | -1 | -1 |



Built for CSE 455 by Derek Zhu and Ruslan Mukhamedvaleev.


### UW CSE 455 Kernel Cooker ⓘ

Kernel: Ridge ▾ Image: Custom Image ▾ Apply Kernel Reset Image

Kernel Matrix ⓘ  
Enter a 3x3 matrix to apply a custom kernel to the image.


|    |    |  |
|----|----|--|
| -1 | -1 |  |
| -1 | 8  |  |
|    |    |  |

**Custom Kernel Settings**  
You can upload your own image here, and customize some other settings.

Fraction: 1  
Custom Image: 

Click to upload or drag and drop  
PNG or JPG (MAX. 450 × 450px)

Done



### UW CSE 455 Kernel Cooker ⓘ

Kernel: Sharpen ▾ Image: Custom Image ▾ Apply Kernel Reset Image

Kernel Matrix ⓘ  
Enter a 3x3 matrix to apply a custom kernel to the image.

|    |    |    |
|----|----|----|
| 0  | -1 | 0  |
| -1 | 5  | -1 |
|    |    |    |

